CLAIMS:

What is claimed is:

1. A method for validating a hardware design, comprising:

applying one of a plurality of transformation rules to simplify a binary decision diagram containing function symbols and variables which represent a hardware design to be validated;

repeating the application of the plurality of transformation rules to the binary decision diagram until no more of the plurality of transformation rules may be applied to the binary decision diagram; and

in response to no more of the plurality of the transformation rules being applicable to the binary decision diagram, determining whether the binary decision diagram has been reduced to a single true value.

2. The method of claim 1, further comprising:

defining a first ordering relation on a set of terms, wherein the terms include function symbols and variables.

3. The method of claim 2, wherein the first ordering relation follows a subterm property.

4. The method of claim 2, wherein the first ordering relation follows a monotonicity property.

5. The method of claim 2, further comprising:

in response to defining the first ordering relation, defining a second ordering relation on a set of equalities, wherein the set of equalities includes equalities between terms ordered by the first ordering relation.

6. The method of claim 1, wherein the plurality of transformation rules includes mapping a node of the form $ite(s=s,H,K)$ into a node of the form H.

7. The method of claim 1, wherein the plurality of transformation rules includes mapping a node of the form $ite(s=t,H,K)$ into a node of the form $ite(t=s,H,K)$ in response to a determination that t is greater than s in an ordering relation having a subterm property and a monotonicity property.

8. The method of claim 1, wherein the plurality of transformation rules includes mapping a node of the form $ite(s=t,H,H)$ into a node of the form H.

9. The method of claim 1, wherein the plurality of transformation rules includes mapping a node of the form $ite(s=t,ite(s=t,H,K),L)$ into a node of the form $ite(s=t,H,L)$.

10. The method of claim 1, wherein the plurality of transformation rules includes mapping a node of the form $ite(s=t,H,ite(s=t,K,L))$ into a node of the form $ite(s=t,H,L)$.

11. The method of claim 1, wherein the plurality of transformation rules includes mapping a node of the form $ite(s_1 = t_1, ite(s_2 = t_2, H, K), L)$ into a node of the form $ite(s_2 = t_2, ite(s_1 = t_1, H, L), ite(s_1 = t_1, K, L))$ in response to a determination that $s_1 = t_1$ is greater than $s_2 = t_2$ according to a pre-determined ordering relation.

12. The method of claim 1, wherein the plurality of transformation rules includes mapping a node of the form $ite(s_1 = t_1, H, ite(s_2 = t_2, K, L))$ into a node of the form $ite(s_2 = t_2, ite(s_1 = t_1, H, K), ite(s_1 = t_1, H, L))$ in response to a determination that $s_1 = t_1$ is greater than $s_2 = t_2$ according to a pre-determined ordering relation.

13. The method of claim 1, wherein the plurality of transformation rules includes mapping a first set of nodes that are true children of a node of the form $ite(s = t, H, K)$ into a second set of nodes that is identical to the first set of nodes except that occurrences of s in the first set of nodes are replaced by t in the second set of nodes.

14. A computer program product in a computer-readable medium for validating a hardware design, comprising functional descriptive material that when executed by a computer, enables the computer to perform acts including:

applying one of a plurality of transformation rules to simplify a binary decision diagram containing function

symbols and variables which represent a hardware design to be validated;

repeating the application of the plurality of transformation rules to the binary decision diagram until no more of the plurality of transformation rules may be applied to the binary decision diagram; and

in response to no more of the plurality of the transformation rules being applicable to the binary decision diagram, determining whether the binary decision diagram has been reduced to a single true value.

15. The computer program product of claim 14, comprising additional functional descriptive material that when executed by the computer, enables the computer to perform additional acts including:

defining a first ordering relation on a set of terms, wherein the terms include function symbols and variables.

16. The computer program product of claim 15, wherein the first ordering relation follows a subterm property.

17. The computer program product of claim 15, wherein the first ordering relation follows a monotonicity property.

18. The computer program product of claim 15, comprising additional functional descriptive material that when executed by the computer, enables the computer to perform additional acts including:

in response to defining the first ordering relation, defining a second ordering relation on a set of equalities, wherein the set of equalities includes equalities between terms ordered by the first ordering relation.

19. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a node of the form $ite(s = s, H, K)$ into a node of the form H.

20. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a node of the form $ite(s = t, H, K)$ into a node of the form $ite(t = s, H, K)$ in response to a determination that t is greater than s in an ordering relation having a subterm property and a monotonicity property.

21. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a node of the form $ite(s = t, H, H)$ into a node of the form H.

22. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a node of the form $ite(s = t, ite(s = t, H, K), L)$ into a node of the form $ite(s = t, H, L)$.

23. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a

node of the form $ite(s=t,H,ite(s=t,K,L))$ into a node of the form $ite(s=t,H,L)$.

24. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a node of the form $ite(s_1=t_1,ite(s_2=t_2,H,K),L)$ into a node of the form $ite(s_2=t_2,ite(s_1=t_1,H,L),ite(s_1=t_1,K,L))$ in response to a determination that $s_1=t_1$ is greater than $s_2=t_2$ according to a pre-determined ordering relation.

25. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a node of the form $ite(s_1=t_1,H,ite(s_2=t_2,K,L))$ into a node of the form $ite(s_2=t_2,ite(s_1=t_1,H,K),ite(s_1=t_1,H,L))$ in response to a determination that $s_1=t_1$ is greater than $s_2=t_2$ according to a pre-determined ordering relation.

26. The computer program product of claim 14, wherein the plurality of transformation rules includes mapping a first set of nodes that are true children of a node of the form $ite(s=t,H,K)$ into a second set of nodes that is identical to the first set of nodes except that occurrences of s in the first set of nodes are replaced by t in the second set of nodes.

27. A data processing system for validating a hardware design, comprising:

a processing unit including at least on processor; memory; and

a set of instructions in the memory,

wherein the processing unit executes the set of instructions to perform acts including:

applying one of a plurality of transformation rules to simplify a binary decision diagram containing function symbols and variables which represent a hardware design to be validated;

repeating the application of the plurality of transformation rules to the binary decision diagram until no more of the plurality of transformation rules may be applied to the binary decision diagram; and

in response to no more of the plurality of the transformation rules being applicable to the binary decision diagram, determining whether the binary decision diagram has been reduced to a single true value.

28. The data processing system of claim 27, wherein the processing unit executes the set of instructions to perform additional acts including:

defining a first ordering relation on a set of terms, wherein the terms include function symbols and variables.

29. The data processing system of claim 28, wherein the first ordering relation follows a subterm property.

30. The data processing system of claim 28, wherein the first ordering relation follows a monotonicity property.

31. The data processing system of claim 28, wherein the processing unit executes the set of instructions to perform additional acts including:

in response to defining the first ordering relation, defining a second ordering relation on a set of equalities, wherein the set of equalities includes equalities between terms ordered by the first ordering relation.